PCT/ SE 99 / 0 2 1 6 7

REC'D 24 JAN 2000

WIPO PCT

4

SE99/2167

PRV

PATENT- OCH REGISTRERINGSVERKET
Patentavdelningen

PATENT- OCH REGISTRERINGSVERKET

Intyg
Certificate

Härmed intygas att bifogade kopior överensstämmer med de handlingar som ursprungligen ingivits till Patent- och registreringsverket i nedannämnda ansökan.

This is to certify that the annexed is a true copy of the documents as originally filed with the Patent- and Registration Office in connection with the following patent application.

(71)	*Sökande* *Applicant (s)*	*Telefonaktiebolaget L M Ericsson, Stockholm SE*
(21)	*Patentansökningsnummer* *Patent application number*	*9804027-2*
(86)	*Ingivningsdatum* *Date of filing*	*1998-11-24*

Stockholm, 2000-01-10

För Patent- och registreringsverket
For the Patent- and Registration Office

Emma Johnsson

Emma Johnsson

Avgift
Fee

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

Ink. t. Patent- och reg.verket

1998 -11- 24

SYSTEM AND METHOD FOR TRANSFERRING LOCALLY HELD INFORMATION IN A COMPUTER COMPLEX

Technical field of the invention

The present invention generally relates to loosely coupled multiprocessor systems and is particularly associated with the task of transferring locally held information between a plurality of processors and a central or administrative processor in such a loosely coupled system.

Specifically, the invention may be applied within the field of mobile telephone systems, whereby a plurality of counter values are to be updated and processed.

Background art

Certain computational problems require extensive amounts of processing power and can not be resolved by using only a single processor because of its inherent limitations in processing speed. Moreover, breakdowns in single processor systems often have dire consequences because the whole system is affected.

These computational problems are typically resolved by means of multiprocessing systems in which more processors work simultaneously on sub-tasks belonging to a problem complex and in which a measure of redundancy is incorporated in the system, thereby rendering the system less sensitive to fault conditions.

Multiprocessor systems are typically divided into two main classes:

A first class, denoted closely coupled systems, refer to systems in which shared memory resources are accessible from a plurality of processors.

A second class, denoted loosely coupled systems, refers to systems in which a plurality of processors associated with local memory resources is interconnected in a network. The individual processors are adapted to obtain information held in the memory associated with other processors by requesting this information from the particular processors in question.

An example of such a loosely coupled system is for instance known from US-A-5 390 316 which deals with the operation and configuration of computer system consoles in a

multi-computer complex. An essential problem addressed in this document is the simultaneous access of and sharing of data among a plurality of users in a computer system complex. The system according to the above document is provided with a mechanism for message delivery and redelivery in a controlled manner through the use of distributed shared memory.

Switching systems for telecommunication networks is another example of systems, which involves controlling a vast number of operations simultaneously.

The Ericsson ™ digital switch AXE 10 ™ is an example of a complete network system which serves to interconnect networks such as ISDN (Integrated Services Digital Network, PSTN (Public Switched Telephone Network, PLMN (Public Land Mobile Network) and a business communications network.

The hardware scheme of the AXE 10 has been sketched in figure 1 and comprises the following elements: A switching hardware section, APT, a number of regional processors, RP, being connected to the switching hardware APT, an administrative processor complex CP, and a set of support processors, SP, these being coupled by means of a regional processor bus, RPB. Each respective processor is associated with local shared memory, which can be addressed by other processors.

The regional processors, RP, control the switching hardware APT and deals with routine tasks in the network, such as performing the individual channel connections in the network and such as performing digital / analogue signal conversion. The administrative processor complex, CP, on the other hand, controls the overall functioning of the digital switch and handles complex decision making tasks, which can be characterised as being mainly of an analytical or administrative nature.

The support processors, SP, handle man machine interface, file management and data communication tasks. For this reason, the support processors are coupled via a LAN (Local Area Network) to a man-machine communication subsystem comprising equipment such as, PC, printer, hard disk and data link.

By means of an operation and management subsystem, OMS (not shown), implemented in central and regional software, operational parameters, such as new subscriber data, exchange data etc. of the system can be updated. This system comprises among other

things a network management system, NMS, which is implemented on the support processors, SP, and which deals with statistical and operational data relating to the performance of the overall system. In addition, a file management system, FMS, based on a mass storage system such as a hard disk is coupled to the LAN.

Under the operational and management system, OMS, a statistics and traffic measurement subsystem, STS (not shown) collects, stores, process and presents measurement data, such as traffic flow and network performance. Also, a charging subsystem, CHS (not shown), responsible for the charging of subscribers calls, is provided under the OMS.

In operation, locally held counters relating to the traffic are read at specific intervals and stored in a measurement database, such that the traffic flow can be monitored and such that detailed information about each charged call can be established. This information is downloaded on the file management system, FMS, for later transferral to external billing centres.

As mentioned above, some of the collected data are typically of a statistical nature and are based on the summation of cumulated values.

Having regard to public land mobile networks, PLMN, which the above mentioned switch is adapted to function in conjunction with, data are in many cases transmitted between and stored at many local base station processors which are associated with the regional processors.

The task of collecting these data becomes complex, when a mobile end station, MES is travelling through more cells, because data have to be transported from the base stations associated with the cells to for instance the file management system, FMS. This activity involves a plurality of administrative messages being transmitted in the network, which activity is taking up capacity.

Consequently, the capacity relating to transmitting messages between end terminals, such as telephone calls being transmitted on the network between regional processors are reduced accordingly.

Summary of the invention

As mentioned above, the present invention is belonging to the class of loosely coupled systems and is particularly directed to a multiprocessor system in which data can be transferred between an administrative processor and a plurality of worker processors.

One object of the present invention is to provide a multiprocessor system and a method therefor for gathering counter values, which are held in respective local worker processors and which are updated or incremented locally under the control of the local worker processor, and for subsequently performing a central updating or accumulation of the gathered counter values in an administrative processor. This object should be performed having regard to substantially limiting or obviating transmitting redundant messages on the network.

This object has been accomplished by the subject matter set forth in claims 1 and 7.

It is another object of the present invention to secure redundancy of data and enable restoring of data to local worker processors in case that one of these should malfunction.

This object has been achieved by the subject matter defined in claims 2 and 8.

It is still another object to obviate the transferral of locally held counter values which have stopped changing.

This object has been accomplished by the subject matter defined in claims 4 and 9.

Further advantages will appear from the remaining claims and the description.

It should be understood that the data held in the worker processors may not exclusively concern counters which are to be incremented but may concern other types of data, such as log files or other data which is accumulated over time.

It should also be noted that many different types of data counters may be involved, and that specific types of counters or data may be assigned or created on a particular processor or deleted.

Specifically, the invention may be applied within the field of mobile telephone systems, although the present invention is not in any way limited to such systems.

The present invention defines a multiprocessor system, which enables information held locally on worker processors to be transferred to an administrative processor while substantially reducing or entirely obviating transmitting redundant messages on the network, thereby providing more capacity for other purposes.

Drawings

Fig. 1 shows a known so called digital switch, which, among other things, can be used in conjunction with a land mobile telephone network.

Fig. 2 shows a schematic illustration of the data-structures being involved for a preferred first embodiment and a second embodiment of the present invention.

Fig. 3 shows a flow-chart of a first routine according to a first preferred embodiment being implemented on the worker processor.

Fig. 4 shows a flow chart of a second routine according to a first preferred embodiment being implemented on the administrative processor.

Fig. 5 shows a handshake diagram relating to the first and second routines shown on figs. 3 and 4.

Fig. 6 shows a flow-chart of a third routine according to a second preferred embodiment being implemented on the worker processor.

Fig. 7 shows a flow chart of a fourth routine according to a second preferred embodiment being implemented on the administrative processor.

Fig. 8 shows a handshake diagram relating to the third and fourth routines shown on figs. 6 and 7.

Ink. t. Patent- och reg.verket

1998 -11- 2 4

Description of a preferred first embodiment of the invention

The present invention is adapted to be implemented in a processor complex whose physical outline resembles the digital switch shown in fig. 1, that is, a processor complex comprising an administrative processor, AP, and a number of worker processors, WP1, WP2, WPn being interconnected by means of a network. In the following description, the term administrative processor corresponds to the term support processor and the term worker processor corresponds to the term regional processor as discussed in connection with the digital switch shown in fig. 1.

It should be understood that the term administrative processor also covers a processor complex, such as a set of processors operating in parallel. It should furthermore be understood that local memory resources, such as random access memory, RAM, are associated with the administrative processor and the respective worker processors.

In fig. 2, a schematic illustration of the information flow and the data-structures involved has been shown. These data-structures shall now be further described with regard to the preferred first embodiment of the invention.

In the memory associated with each worker processor, WP1 and WP2 there is stored a number of values, which may continuously be updated or incremented according to certain programs run on the respective worker processor.

The values are arranged in so-called probes. A probe is a data-structure, which is used in the worker processors and in the administrative processor. The probe data-structure may be arranged so as to comprise the following entities:

probe-id; probe value (I)

The probe-id identifies the type of probe.

The probe value associated with the probe-id can be an integer value for use as a counter, which for instance is adapted to be incremented, or it may be any other value, which is to be changed over time. It can also be a file, such as a log file referring to a cumulative list of events.

On each worker processor, several types of probes or probe-id's may be assigned. In the course of time, new probes may be assigned, and still other probes may be deleted, i.e. cease to exist.

On the administrative processor duplicates of the values corresponding to the respective probes on the worker processors are established during regular update routines.

A measurement instance is a data-structure, which is used, in the administrative processor, AP. The measurement instance data-structure may comprise the following entities:

probe-id; measurement value; worker registration; probe-links (II)

The probe-id identifies the measurement instance and corresponds to a certain type of probe-id.

The measurement value could for instance relate to the cumulative value of the probe values relating to a given type of probe, i.e. probe-id.

The worker registration is a list of worker processors, on which a given type of probe currently exists.

The probe link corresponding to a given probe type comprises a link of all the probes which currently exist, whereby each individual probe link could comprise the following entities:

processor-id and probe value (III)

In the above entities, the processor-id identifies the respective worker processor and the probe value represents a copy of the corresponding probe value on the respective worker processor.

In fig. 2, by way of example, respective probes, I, existing on the respective worker processors, WP1 and WP2, have been illustrated. On WP1 probe types a - c; e exist, while probe types a; d - f exist on WP2. Probes, I, for which no value has been assigned, have been shown by means of dotted lines.

Moreover, measurement instances, II, have been indicated on the administrative processor, AP, comprising links, III, of probes, I, and it has been indicated by way of example, that a measurement instance, II, comprises probe links whose individual probe values correspond to probe values on the respective registered worker processors, WP1 and WP2. On the administrative processor, all probe types exist.

For illustrative purposes, only probe types d - i have been shown on the administrative processor AP, but it should be understood that the list would also include at least probe types a - c and that the list could be much longer.

Function of first preferred embodiment

The functioning of the method for the transferral of values between the respective worker processor and the administrative processor shall now be explained with reference to the routines shown in fig. 3 and 4 and the handshake diagram shown in fig. 5.

In this embodiment, the administrative processor is assumed being coupled to an external processor as explained above, but it should be understood that the control tasks of the external processor might be implemented on the administrative processor instead.

According to local routines implemented on the respective worker processors as mentioned above, each respective worker processor performs computations and registers events which cause the individual probe values being stored on the local memory associated with the worker processor to be updated.

This locally executed routine, also denoted as the first notification routine, running on the respective worker processors has been shown in fig. 3.

When an event triggers a new probe value to be created or assigned locally, step 1, a first routine in the worker processor ensures that the administrative processor is notified that a probe is created on the respective worker processor, step 2. The administrative processor then registers the given type of probe as existing on the worker processor and requests the current probe value from the worker processor.

Subsequently, the worker processor is continuously carrying out possible changes in probe values, step 3.

When an event occurs in the local worker processors which affects the probe in question to be deleted, step 4, the worker carries on to step 5, in which the worker processor requests the administrative processor to de-register the probe in question.

The administrative processor on the other hand, continuously keeps a record of which probe types currently exist on the various worker processors in the worker registration mentioned above.

Parallel with the above routine, or routines - if all worker processors are taken into account - a first updating routine is executed on the administrative processor. This first updating routine has been shown in fig. 4.

When a need to update or monitor a certain type of probe value occurs, step 1, a request for a measurement is made, step 2. That is, the probe values corresponding to a certain probe type is requested from those workers, which have been registered according to the first routine mentioned above, steps 3 and 4. Subsequently, the probe values in question are delivered, step 5.

Consequently, when a control system, such as a network management system, issues a command for an update of the measurement values stored on the administrative processor, the administrative processor only access those processors on which the probe in question exists and requests only those probe values. This situation has been shown in the upper part of fig. 5, c.f. items 1) - 4).

Then the current accumulative probe values are stored on the administrative processor and the corresponding measurement values are calculated and stored in the measurement instance (II) on the administrative processor, c.f. item 5).

Subsequently, a response is issued from the administrative processor to the network management system, 6).

This mechanism has the effect that considerable improvements with regard to avoiding communicating redundant messages on the network are accomplished.

In fig. 5, it is shown that probes may be requested de-registered on the administrative processor, 7), and subsequently registered, 8).

The above implementation also accomplishes that probe values can be written back to the worker processors in case that one or more worker processor should loose the content of the local memory. This has been shown in the lower part of fig. 5, where a restart, c.f. item 9), has been performed on the worker processor, causing all probe values and registrations to be deleted.

During this re-write procedure, all probe values are transferred back or re-written to the worker processor. Consequently, only recent updated values will be lost.

Second embodiment

Some types of applications may have a burst like character of changing probe values interrupted by longer periods of inactivity.

The second embodiment of the present invention is a further development of the first embodiment and seeks to optimise the performance with regard to the above pattern of burst like changing values.

The data-structures used in this implementation are similar to the above, except that the data-structure relating to the probe links stored in the memory associated with the administrative processor and the data-structure on the worker processor are modified. Reference should be made to fig. 2.

The data-structure on the worker processor is having the following configuration:

probe id; probe value; and timer value (IV)

The timer value is simply a counter which is incremented according to the time and which is adapted to be compared with a predetermined time-out value.

It should be noted that the probe data-structure IV above corresponds to probe I as defined according to the first embodiment of the invention

The data-structure on the administrative processor, the probe link, is having the following configuration according to the second embodiment:

processor id; probe state; and probe value (V)

The probe state can assert either an active or in-active state and serves to indicate whether the timer value has exceeded the time-out value without the probe value having been changed. This condition serves to indicate whether it is unlikely that the probe value in question on a certain worker processor will change again.

It should be noted that the probe data-structure V above corresponds to probe III as defined according to the first embodiment of the invention

The measurement instance according to the second embodiment of the invention is identical to the measurement instance II defined according to the first embodiment.

<u>Function of second embodiment</u>

A second notification routine executed on the local worker processors has been shown in fig. 6. This routine corresponds largely to the first notification routine shown on fig. 3 with the exception that a timeout feature has been incorporated in step 3.

In step 3, it is established whether the time since last change in probe value exceeds a certain predetermined interval. If this is the case, the state of the probe is marked as in-active and the current probe value is transferred to the administrative processor. The above transferral may involve that the administrative processor is reading the current value, when the administrative processor receives the information that the probe in question has been marked in-active.

Should changes occur again within the predetermined interval, the state of the probe is marked active.

In line with the first notification routine, it is examined whether the probe should be deleted and if so, a request for de-registration of the corresponding probe on the particular worker processor is issued to the administrative processor. These measures are carried out in steps 4 and 5.

Parallel to the second notification routine executed on the respective worker processors, a second updating routine, shown in fig. 7, is executed on the administrative processor.

In the second updating routine, which otherwise corresponds closely to the first updating routine, probe values are only transferred from those probes which are both registered and active. This is undertaken in step 3.

Consequently, probe values, which have not changed within the predetermined interval, will not be transferred to the administrative processor.

For probes, which change in a burst like manner the second notification and updating routines have the effect that the traffic relating to transferring probe values, which have not changed for a predetermined period of time - and which are therefore not deemed likely to change in the near future - are not acquired. This mechanism accomplishes savings with regard to issuing administrative messages on the network.

In fig. 8, a handshake diagram pertaining to the second notification and updating routines has been shown.

This figure exemplifies the communication between two worker processors, WP1 and WP2, and an administrative processor, AP, and a network management system, NMS.

In this example, an event, c.f. item 1), causes probes, corresponding to a certain probe type to be created and values to be stored at the two worker processors. Subsequent requests are issued to the administrative processor so as to register the probes, 2). Both probes are active because, the changes have been performed within the predetermined interval for marking them active.

At a later instance, an event, 3), in the network management system, NMS, triggers the administrative processor AP to retrieve the respective active probe values from the first

and the second worker processor, and subsequently store these values on the administrative processor, 4) - 9).

A measurement value is subsequently calculated, 10), on the basis of these probe values and transferred to the network management system.

At a later point in time the probe value on the first worker processor WP1 experiences a timeout; i.e. the probe value has not been updated within the predetermined interval. Consequently, the first worker processor WP1 issues a request to the administrative processor so as to mark the probe in-active, 11), and the current probe value is then transferred to the administrative processor, in which it is stored, 12).

For illustrative purposes, at a still later point in time, the first worker processor experiences a down-break, which causes probe values to be lost on the worker processor, 13 - 15).

A re-write procedure is then undertaken corresponding to the re-write procedure explained in connection with fig. 5. This involves that the stored probe value transferred to the worker processor WP1 and the probe is marked active, 15) - 16).

The worker processor is again ready to carry out changes in values, which for instance could be incremented. Thus probe values could be performed immediately, and subsequently be incremented with the probe value passed from the administrative processor, AP.

Ink. t. Patent- och reg.verket

1998 -11- 2 4

Patent claims

1. A multiprocessor system, comprising an administrative processor and a plurality of worker processors comprising CPU and memory, the processors being interconnected by means of a network, the administrative processor and the worker processor being adapted for storing values in data-structures denoted probes whereby

the administrative processor is adapted to keep a record of whether a certain type of probe exists on a particular worker processor, and whereby

each worker processor is adapted for creating and deleting probe types, and changing or incrementing corresponding probe values held on the respective worker processors, and

in case a probe type is created or deleted on a worker processor, the worker processor being adapted for initiating a notification routine in which the administrative processor is notified about the corresponding creation or deletion of the probe type on the particular worker processor;

the administrative processor is being adapted for initiating an updating routine, in which the administrative processor is acquiring a certain type of probe value from only those particular workers, for which the particular probe type exists, whereby the administrative processor is storing the acquired probe values in a data-structure on the administrative processor.

2. A multiprocessor system according to claim 1, in which the administrative processor is adapted to run a re-write procedure to a particular worker processor, in which probe values are being re-written to the particular worker processor.

3. The multiprocessor system according to claim 1 or 2, wherein the administrative processor is coupled to an external processor by means of a second network, the external processor being able to trigger the updating routine of the administrative processor.

4. A multiprocessor system according to the preceding claims, whereby

the administrative processor is recording the state of the probe as active or in-active, the active state being indicative of the probe value having changed within a predetermined period of time since the preceding change, and the in-active state being indicative of the probe value having remained unchanged within the predetermined period, and whereby

the administrative processor is adapted to carry out an updating routine in which the administrative processor is acquiring a certain type of probe value from only those worker processors on which the particular probe type exists in case the respective probe adopts an active state.

5. A multiprocessor system according to claim 4, whereby the worker processor is adapted to carry out a notification routine in which the administrative processor is notified about the probe state and the probe value, when the probe adopts an in-active state.

6. A multiprocessor system, according to any of the preceding claims, in which the administrative processor and the respective worker processors form part of a network switching system.

7. A method for transferring values being held locally in a computer complex comprising worker processors and an administrative processor, the processors being coupled by means of a network, the type of values being stored in data-structures denoted probes, characterised in that

each individual worker processor performs a notification routine, whereby the administrative processor is requested to register if probe types are being created and to register if probe types are being deleted,

the administrative processor performs an updating routine whereby values corresponding to a certain type of probe are acquired by the administrative processor from only those worker processors on which the probe in question exists.

8. A method for transferring locally held values in a computer complex, according to claim 7, whereby

the administrative processor is adapted to run a re-write procedure to a particular worker processor, in which probe values are being re-written to the particular worker processor.

9. A method for transferring locally held values in a computer complex, according to claim 7 or 8, whereby,

the notification routine furthermore provides for marking the state of the probe as active if the particular probe value has changed within a predetermined period of time, and otherwise marking the probe as in-active, and transferring the probe value to the administrative processor in case the probe becomes in-active,

the updating routine furthermore providing that probe values are acquired from only those particular worker processors for which the probe in question exist and are active

Abstract

A multiprocessor system comprising an administrative processor and a plurality of worker processors coupled in a network have been provided, whereby routines are implemented for the transferral of locally held counter values between worker processors and the administrative processor achieving a substantial reduction of the transmittal of redundant administrative messages on the network and thereby enhancing the overall user capacity of the network.

Fig. 3 + 4.

Ink. t. Patent- och reg.verket

1998 -11- 24

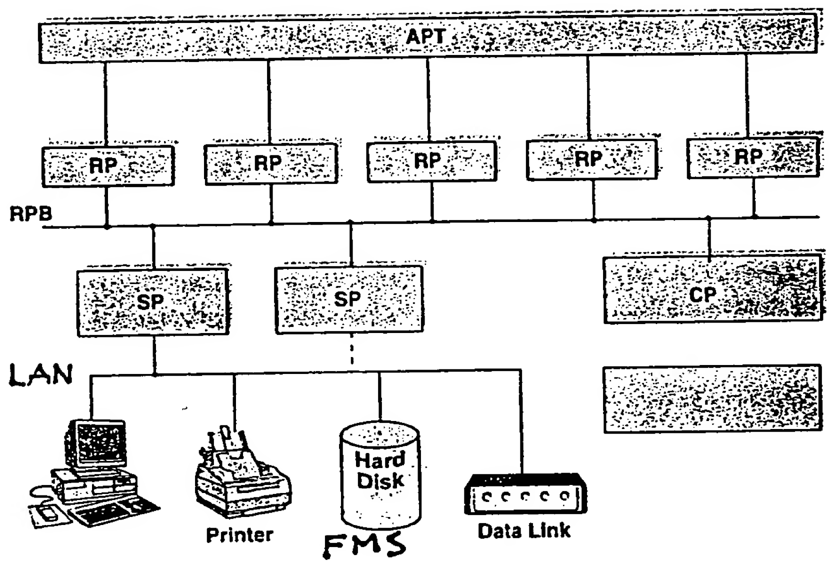

FIG. 1

Ink. t. Patent- och reg.verket

1998 -11- 24

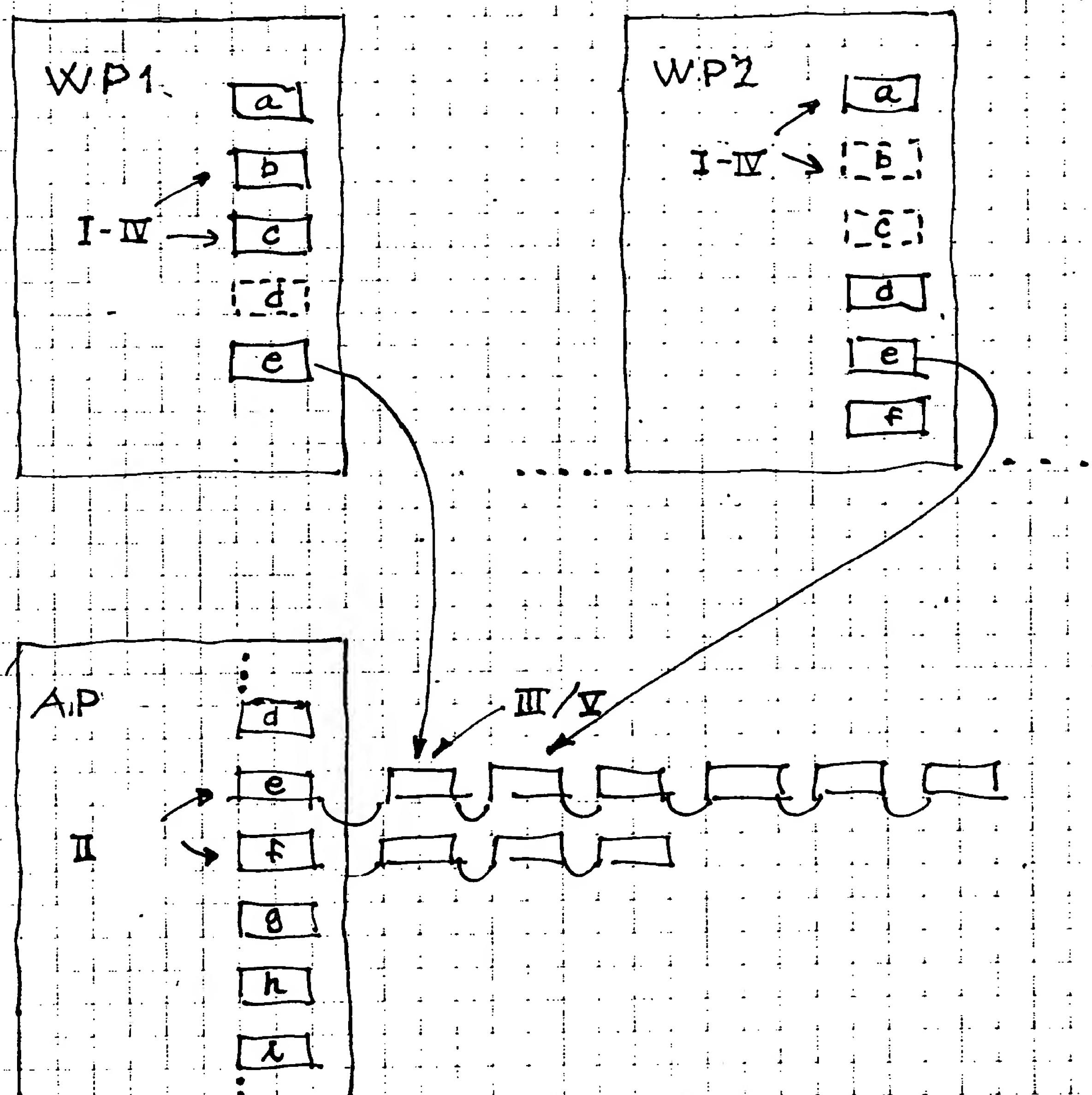

FIG. 2

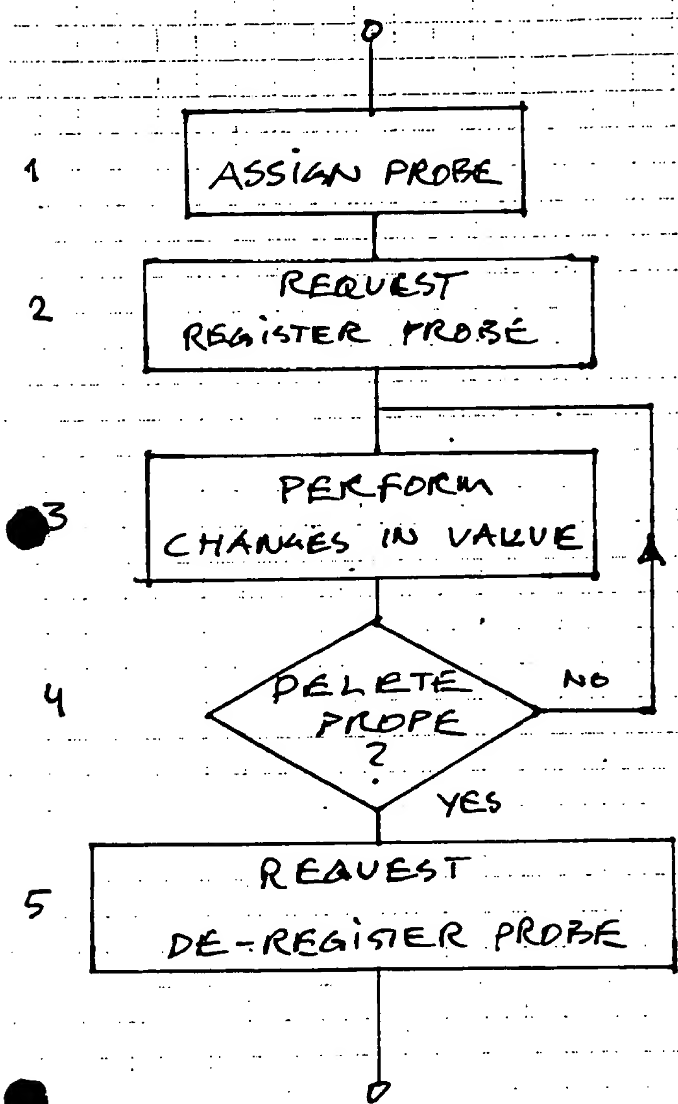
1
ASSIGN PROBE
2
REQUEST
REGISTER PROBE
3
PERFORM
CHANGES IN VALUE
4
DELETE
PROPE
?
NO
YES
5
REQUEST
DE-REGISTER PROBE

FIG. 3

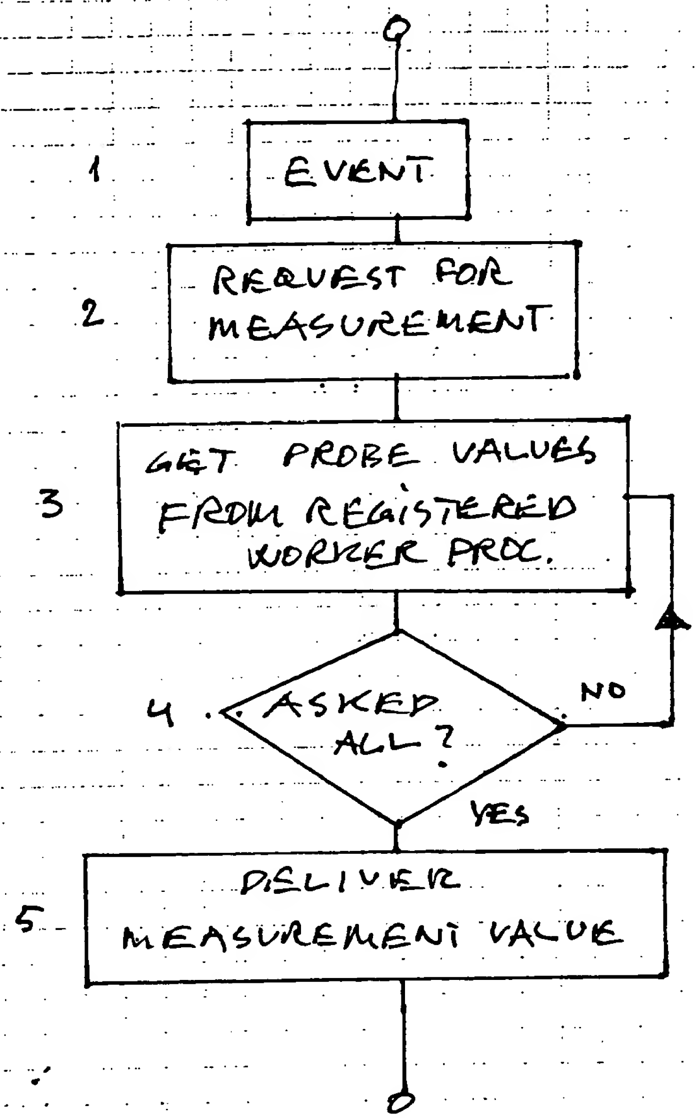
1
EVENT
2
REQUEST FOR
MEASUREMENT
3
GET PROBE VALUES
FROM REGISTERED
WORKER PROC.
4
ASKED
ALL ?
NO
YES
5
DELIVER
MEASUREMENT VALUE

FIG. 4

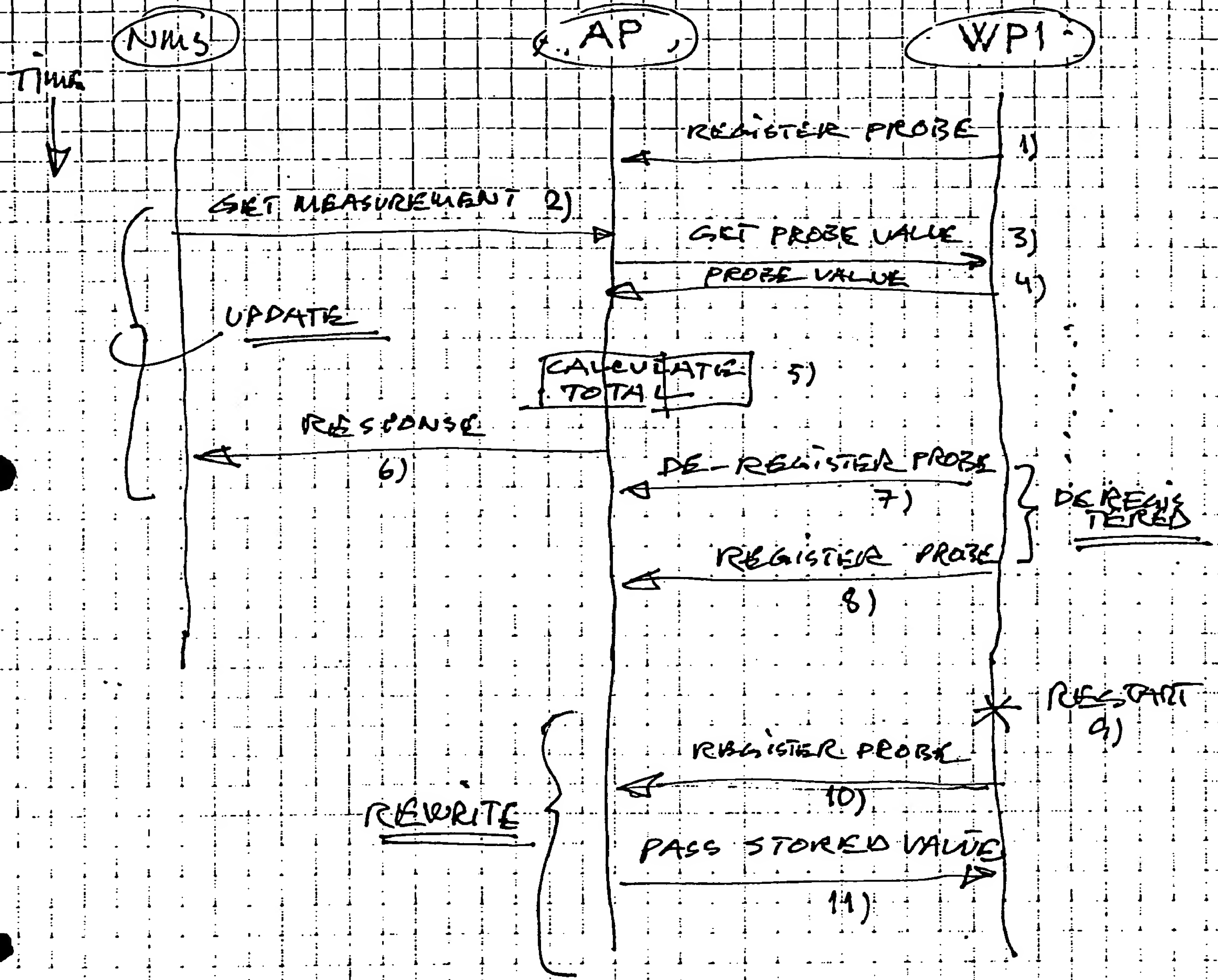
NMS
AP
WP1
TIME
REGISTER PROBE
1)
GET MEASUREMENT 2)
GET PROBE VALUE
3)
PROBE VALUE
4)
UPDATE
CALCULATE TOTAL
5)
RESPONSE
6)
DE-REGISTER PROBE
7)
DEREGISTERED
REGISTER PROBE
8)
RESTART
9)
REGISTER PROBE
10)
REWRITE
PASS STORED VALUE
11)

FIG. 5

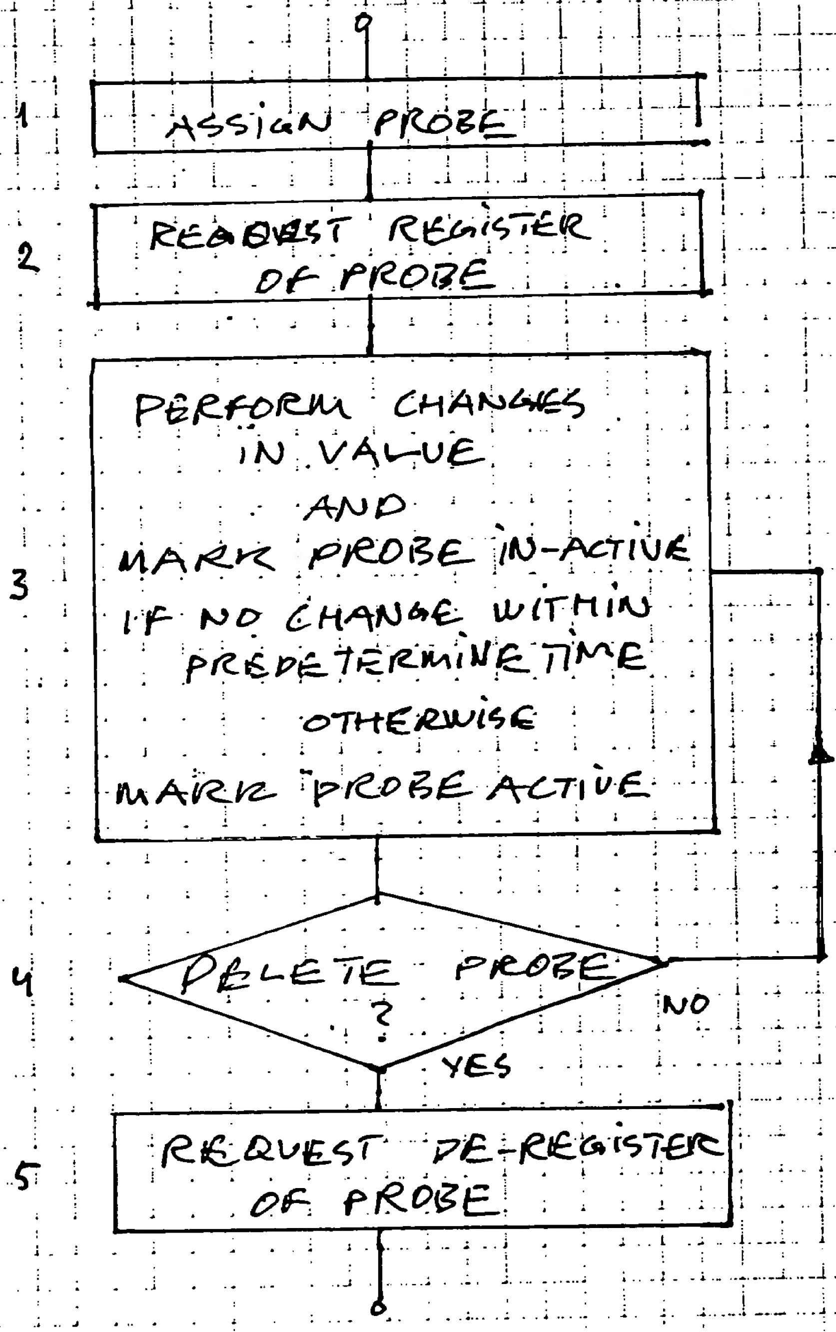
1
ASSIGN PROBE
2
REQUEST REGISTER OF PROBE
3
PERFORM CHANGES IN VALUE
AND
MARK PROBE IN-ACTIVE IF NO CHANGE WITHIN PREDETERMINE TIME
OTHERWISE
MARK PROBE ACTIVE
4
DELETE PROBE ?
NO
YES
5
REQUEST DE-REGISTER OF PROBE

FIG. 6

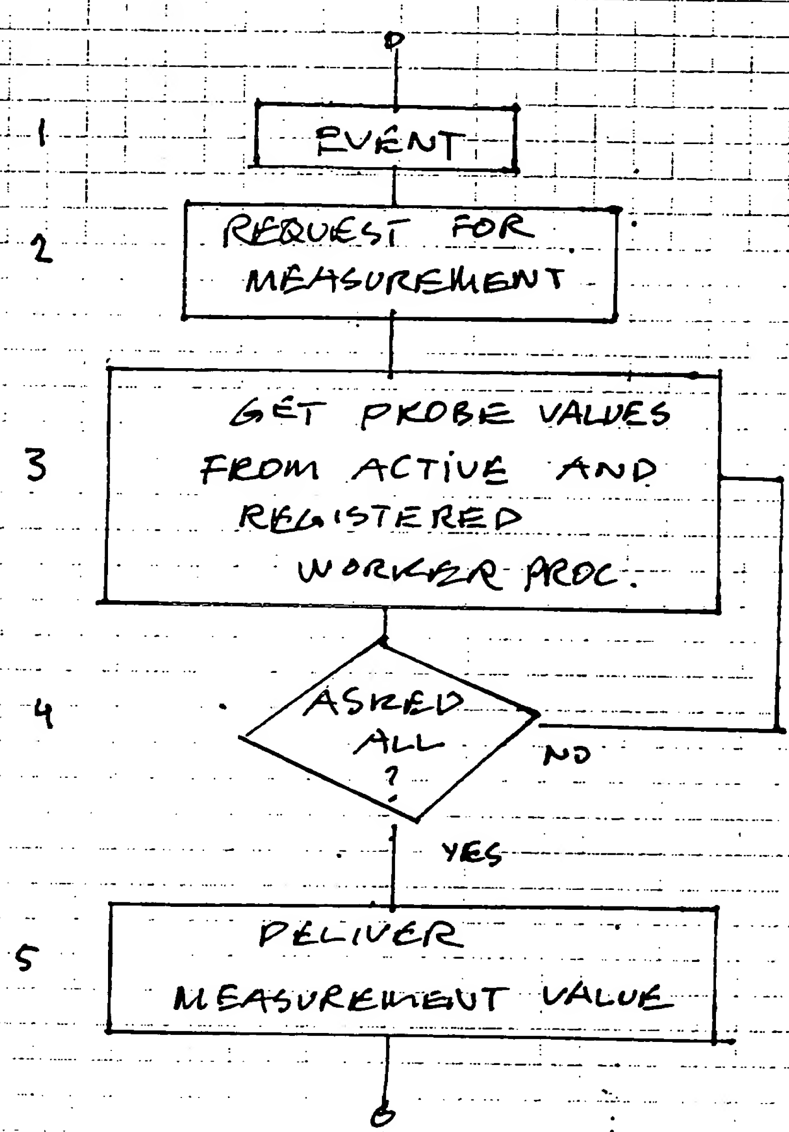
1
EVENT
2
REQUEST FOR MEASUREMENT
3
GET PROBE VALUES FROM ACTIVE AND REGISTERED WORKER PROC.
4
ASKED ALL ?
NO
YES
5
DELIVER MEASUREMENT VALUE

FIG. 7

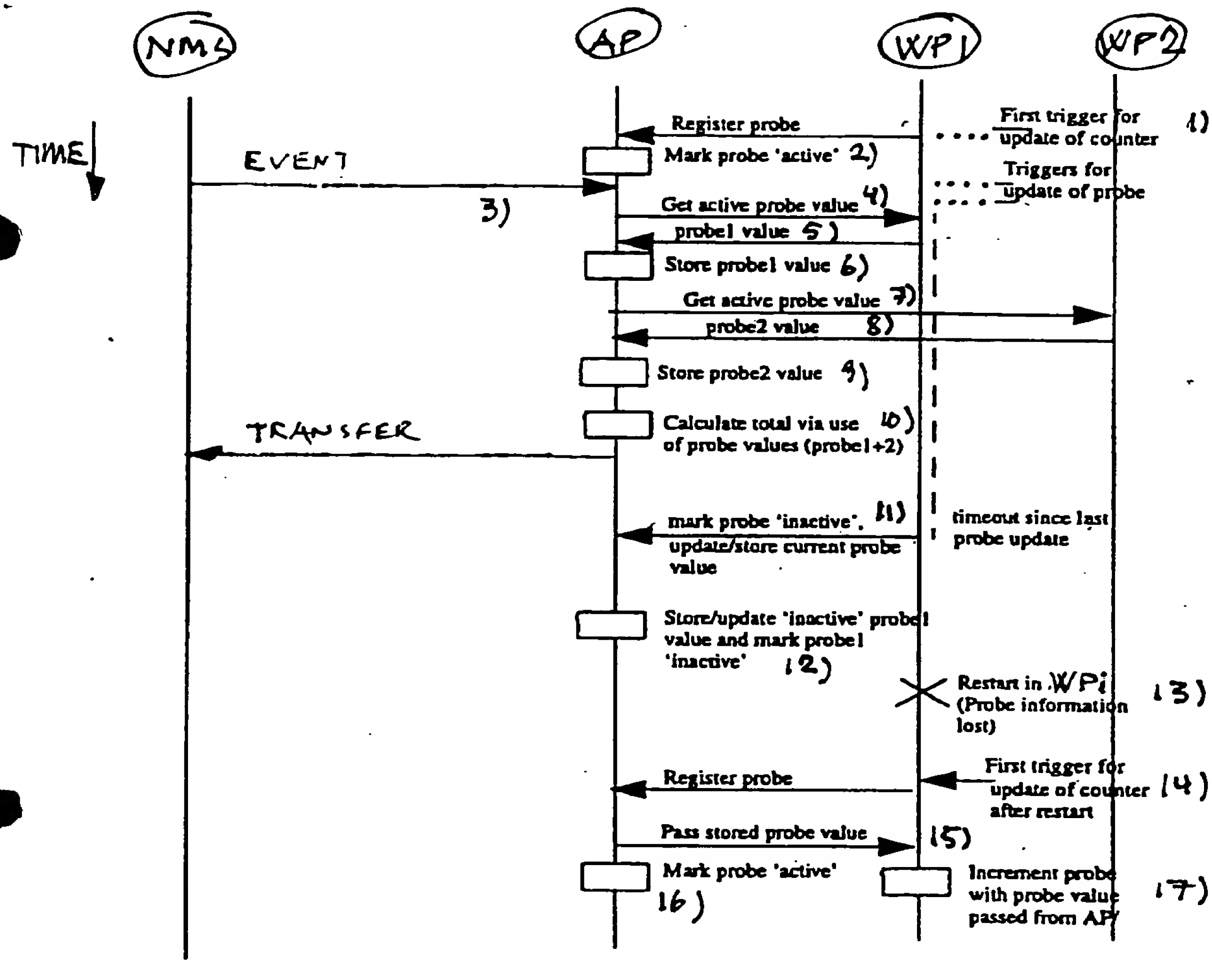
NMS
AP
WP1
WP2
TIME
EVENT
3)
Register probe
First trigger for update of counter 1)
Mark probe 'active' 2)
Triggers for update of probe
Get active probe value 4)
probe1 value 5)
Store probe1 value 6)
Get active probe value 7)
probe2 value 8)
Store probe2 value 9)
TRANSFER
Calculate total via use of probe values (probe1+2) 10)
mark probe 'inactive'. update/store current probe value 11)
timeout since last probe update
Store/update 'inactive' probe1 value and mark probe1 'inactive' 12)
Restart in WP1 (Probe information lost) 13)
Register probe
First trigger for update of counter after restart 14)
Pass stored probe value 15)
Mark probe 'active' 16)
Increment probe with probe value passed from AP 17)

FIG. 8